REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-03-

0382

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
15-08-2003	Final Project Summary	01-02-2001 - 30-06-2003

4. TITLE AND SUBTITLE
Specification-Based Approaches for Information Assurance:

Final Status Report

5a. CONTRACT NUMBER

5b. GRANT NUMBER
F49620-01-1-0332

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Daniel C. DuVarney and R. Sekar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Computer Science
Stony Brook University
Stony Brook NY 11794-4400

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
4015 Wilson Boulevard, Room 713
Arlington, VA 22203-1954

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

20031006 053

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The primary goal of this project is to develop a training program for a CIPIA fellow in the context of our research program outlined above. The fellow's research will center on predicting, preventing, monitoring and responding to intruder attacks. The research/training program is designed to exploit the PI's unique combination of qualifications in order to attract researchers that are currently working in areas other than information assurance. It seeds to provide them attractive opportunities that will enhance the likelihood of continued research by the fellow in information assurance, even after the conclusion of this project.
The CIPIA Fellowship was awarded to Dan DuVarney, who was supported from February 2002 through June 2003. During his tenure as a CIPIA Fellow, Dan was a key contributor in several research projects, including; development of benign software mutations, collaboration with the PI on development of model-extraction techniques for the Model-Carrying Code, and model extraction of communications protocols from systems code.

15. SUBJECT TERMS
Information assurance, formal methods, program security, mobile code security.

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER (include area code)
Unclassified	Unclassified	Unclassified		7	R. Sekar (631) 632-5758

Specification-Based Approaches for Information Assurance
Final Status Report, August 15, 2003.

1 Project Overview

Networked information systems are playing increasingly important roles in our infrastructures for critical services such as commerce, banking and telecommunication. Existing techniques for protecting such systems against intruder attacks are based on a *reactive* approach that does not offer adequate protection against previously unknown attacks. Typical reactive solutions, such as upgrading to a patched version of a software, are short-lived, surviving only until the next attacker identifies an alternative. System administrators are thus in a constant struggle to stay a step ahead of a vast army of resourceful hackers.

Through DoD and NSF sponsored research, we are developing a *proactive* approach that enables the system to anticipate potential problems and protect itself against them. Some of the key benefits of our approach are:

- *prevention and/or damage containment*, as opposed to previous research that was mainly concerned with post-attack detection.
- *accurate detection of known and unknown attacks,* while maintaining a low false-positive rate.
- *applicability to COTS software and systems,* where software is written in multiple languages and runs on multiple platforms, and moreover, the source code is not available.
- *protection against a variety of threats within a single framework,* including threats posed by errors in server or client applications, operating system software, network protocols, and untrusted mobile code.

The primary goal of this project is to develop a training program for a CIPIA fellow in the context of our research program outlined above. The fellow's research will center on predicting, preventing, monitoring and responding to intruder attacks. The research/training program is designed to exploit the PI's unique combination of qualifications in order to attract researchers that are currently working in areas other than information assurance. It seeks to provide them attractive opportunities that will enhance the likelihood of continued research by the fellow in information assurance, even after the conclusion of this project.

At the time the proposal was submitted, it was anticipated that the CIPIA fellow would conduct his/her research and training at the Secure Systems Research Laboratory at SUNY, Stony Brook. The fellow's original research background was expected to be in one or more of the following areas: programming languages, compilers and formal methods. Through the training, the fellow was expected to make a transition to research in CIP/IA, focusing on specification-based approaches for intrusion detection and response.

2 Final Status

The CIPIA Fellowship was awarded to Dan DuVarney, who was supported from February 2002 through June 2003. Dan completed a Ph.D. degree in Computer Science at North Carolina State University. His thesis research at NCSU focused on the development of practical model extraction techniques for C programs [4], which fit very well with ongoing projects at Stony Brook on modeling program behaviors, and verifying the consistency of these models with user-specified security policies. In particular, the Model-Carrying Code (MCC) project (supported by a University Research Initiative grant from ONR) [10, 15, 14], the specification-based intrusion detection/response projects (supported by two grants from NSF and two earlier grants from DARPA) [11, 3, 13, 8, 9], and the model-checking based vulnerability analysis projects (supported by an NSF ITR award) [7, 6] at Stony Brook are all closely aligned with Dan's research background.

During his tenure as a CIPIA Fellow, Dan was a key contributor in several research projects. In addition, he conceived several new research projects as well, some of which have already produced significant results. A quick summary of his work is as follows:

- Development of *benign software mutations*, a novel approach that provides a *systematic defense* that is effective across the classes of buffer-overflow attacks that are known today, and likely to be discovered in the future. The results of this work were recently published [1] at the USENIX Security Symposium.
- Collaboration with the PI on development of model-extraction techniques for the MCC project. This work resulted in a paper that has been accepted for publication [12] at the ACM symposium on operating system principles (SOSP).
- Model extraction of communications protocols from systems code, resulting in a paper which has been published [5] at FORTE, a leading conference in formal methods.
- Development of ideas for the "next generation" of benign software mutations, resulting in a paper [2] that was presented at the New Security Paradigms Workshop.
- Development of a research project (in collaboration with Prof. Sekar) studying the use of source-code transformations to efficiently detect all memory errors in C programs, resulting in a proposal submitted to NSF.
- Development of a research project (in collaboration with researchers at SUNY and NCSU) on model checking of C programs to verify safety and security properties, resulting in a proposal submitted to NSF.
- Development of a research project (in collaboration with Prof. Sekar) investigating the use of benign software mutations to combat self-propagating attacks on the Internet, resulting in a proposal submitted to AFOSR.
- Collaboration with 3 Ph.D. students at SUNY and one M.S. student at NCSU, in the areas of information-flow analysis, program transformations for memory safety, and binary code analysis, and static detection of buffer overflows.
- Development of novel on-the-fly minimization techniques for model extraction.
- Training efforts in Computer Security, through directed readings of the current research literature, books, and participation in research seminars and workshops.

2.1 Model-Carrying Code

Dan's work on model extraction is a critical part of the MCC project. MCC is a new approach for providing secure mobile code, in which a model of a program's behavior (in the form of an extended finite-state machine) is distributed along with the code for the program. The model can be efficiently checked against the code to determine that it is a faithful abstraction of the program's behavior. Once the model is verified, the local security policy can be compared with the model to see if the program should be run. Additionally, the model can be used as a basis for runtime monitoring of the mobile code to ensure that its behavior never violates the model.

Dan is in the process of adapting his model-extraction system to meet the requirements of the MCC project. He has changed the model generation process to generate push-down automata, allowing infinite-state programs to be easily modeled, and has re-targeted the back-end to generate models usable with the XMC model checker that is used in the MCC project. He is currently working on new program analysis techniques to extract data-flow information from the programs and incorporate this information into the models. This would allow us to capture important security-relevant information in the models, e.g., what files are being opened by programs, how they may be related to the parameters passed to the program, etc. The analysis is based on a new abstraction technique that uses symbolic representation of values. The new abstraction approach is automatic in the sense that user-input is not required to perform the abstraction. Additionally, the use of symbolic abstractions allows the easy detection of relationships between program variables (particularly formal arguments to functions), while preserving needed information about constant values (such as constant prefixes to strings and vital integer constants). Both of these are significant departures from earlier work.

2.2 Student Collaboration

Dan has been collaborating with 3 students at SUNY and one student at NCSU. First, he has been working with a Ph.D. student who is developing a practical technique for quantifying information-flow through a program. To date, they have produced an abstract interpreter for a mini-language which quantifies the information (in terms of the number of bits) that flow from a given program variable to each point in the program text. The work is leveraging Dan's thesis work, which developed an abstract interpreter for C programs, once the interpretation technique is fully-specified, the existing interpreter can be adapted to form a tool for quantifying information-flow.

Second, Dan is actively collaborating with a Ph.D. student who is developing C source-code transformation techniques for memory safety. The goal of this work is to develop a source-code transformation which results in a memory-safe program (in which all memory leaks, dangling pointers, temporal memory errors, etc. are at least result in a runtime exception), and introduces minimal runtime overhead. As compared with previous research on this problem, our technique is designed to work without requiring any changes to source code of existing programs, including programs that cast pointers in arbitrary ways. Moreover, our technique does not rely on garbage collection. To date, Dan has made some important contributions, including a timestamp-based optimization that significantly reduces the runtime overhead. Dan also helped direct the preparation of the student for his Research Proficiency Exam, which the student recently passed.

Third, Dan is interacting with a Ph.D. student who is developing tools for analyzing and transforming binary code. The primary goals of this work are to provide tools for runtime monitoring, static model extraction, and binary program transformation for security. This work also dovetails with Dan's work on benign mutations (section 2.4). This collaboration has already resulted in one published paper [1], and another accepted for publication [2].

Finally, Dan collaborated with an M.S. student at NCSU who extended Dan's thesis work to statically detect buffer overflows. Dan has specified new abstractions which will enable overflows to be more easily detected and the student is implementing them. Dan was appointed as an adjunct assistant professor at NCSU so that he could serve as a member of the student's Thesis Committee; the student recently successfully defended his thesis. A paper on the buffer overflow detection technique is in preparation, and should be submitted for publication in the next two weeks.

2.3 Communications Protocol Extraction

Dan has continued to develop the model-extraction system from his thesis research, and improved the results from his thesis. He successfully extracted models of the behavior of the GNU I-protocol, illustrating differences in behavior between two versions (the earlier version of the I-protocol had a bug which resulted in livelock). Dan presented a paper on this work this November at the International Conference on Formal Techniques for Networked and Distributed Systems, which is one of the premier conferences in formal methods.

Since then, Dan has developed a new method for minimizing models on-the-fly as they are constructed. The method is based the identification of *nearly-closed* subgraphs during model construction. These nearly-closed subgraphs have the desirable property that they can be reduced by using the weak-bisimulation equivalence relation, which produces the largest savings in the number of states possible for the type of model used by C Wolf [5] and similar tools. The near-term goal for this work is to use the improved minimization technique to extract larger and more comprehensive models of the I-protocol, and submit the results for publication.

2.4 Benign Mutations

Dan has also been developing a new approach for securing large-scale systems such as the Internet against rapid spread of viruses and worms. This approach is based on the ecosystems metaphor, in which a network of computer systems is viewed as a collection of biological organisms. The text of the programs running on these systems is analogous to the DNA of the organisms, particularly in the sense that a "mono-culture" in which all systems run the same code is vulnerable to an "epidemic" caused by a single computer virus or other attack in the same way that a real monoculture is vulnerable to being decimated by a real virus.

Sticking with the metaphor, the question is, is there some way to achieve a "biodiversity" of computer systems? It's not practical to require that each computer run a different operating system and/or set of application programs. However, one possible way in which diversity could be achieved is via a tool which randomly mutates binary code in a benign fashion. The term "benign" refers to the fact that the mutations don't affect the user-observable behavior of the program. Instead the mutations involve shuffling the memory usage of the program, which will cause a buffer overflow attack that targets one particular memory layout to fail for all other layouts.

This approach will allow each system on a network to run the same software, while reducing the vulnerability of the network to worms, viruses and intrusions. If each system mutates its software randomly, then, ideally, an attack which succeeds one system will fail on the vast majority of the others.

Through collaboration with other Security Lab members, Dan has developed a concrete application of the mutation idea, called *address obfuscation.* In this approach, a program is transformed so that it pro-actively mutates itself each time it is run. A preliminary analysis of effectiveness and performance was undertaken, we recently presented at the 2003 USENIX Security Symposium [1].

Future plans for this work are to continue development of the address obfuscation transformation tool, pursuing two basic avenues of research. The first is binary-only transformations, which are useful in the case where source code is not available. A paper describing these transformations and how they can be practically achieved has been accepted for publication at the 2003 New Security Paradigms Workshop [2]. The second are source-level transformations, which are easier to perform, but require availability of source code. Additionally, Dan plans on pursuing the more general idea of benign mutations to identify broader applications of the idea.

2.5 Education

Dan has also been actively educating himself in the requisite background knowledge of a security researcher. He has read numerous recent and classic papers, as well as large sections of several books. He has attended the weekly security research seminar at SUNY, and presented results there several times. Dan has attended the AFOSR Workshop on Systems and Information Fusion, the FORTE/PSTV conference, the Summer 2003 ONR MURI PI meeting, and the 2003 Usenix Security Symposium.

3 Future Plans

Dan's plans to continue as a Postdoctoral Research Associate at SUNY at least until the end of this year. The primary goal for the remainder of his term here at SUNY is to complete the projects described in section 2. Important milestones include:

- Improvement of the current source-level model-extraction tool for the MCC project.
- Publishing a paper on runtime enforcement of information-flow policies.
- Developing static analysis techniques to improve the performance of the memory-safe program transformation project.
- Completion of the on-the-fly bisimulation-based minimization work, with an experimental paper published.
- Continuing to gain expertise in the area of Computer Security.

Dan is still investigating career options for once he leaves Stony Brook. His goal is to continue research in the area of information assurance, at either an industrial or academic research institution.

References

[1] BHATKAR, S., DUVARNEY, D. C., AND SEKAR, R. Address obfuscation: an efficient approach for combating a broad-range of memory-error exploits. In *Proceedings of the 12th Usenix Security Symposium* (Washington, D.C., August 2003), Usenix Association.

[2] BHATKAR, S., DUVARNEY, D. C., AND VENKATAKRISHNAN, V. N. Self: a transparent security extension for elf binaries. In *Proceedings of the 2003 New Security Paradigms Workshop* (August 2003).

[3] BOWEN, T., CHEE, D., SEGAL, M., SEKAR, R., UPPULURI, P., AND SHANBAG, T. Building survivable systems: An integrated approach based on intrusion detection and confinement. In *Darpa Information Security Symposium* (2000).

[4] DUVARNEY, D. C. *Abstraction-Based Generation of Finite State Models from C Programs*. PhD thesis, North Carolina State University, 2002.

[5] DUVARNEY, D. C., AND IYER, S. P. C wolf — a toolset for extracting models from c programs. In *International Conference on Formal Techniques for Networked and Distributed Systems (FORTE)* (Houston, TX, USA, November 2002), D. Peled and M. Vardi, Eds., LNCS, Springer Verlag.

[6] RAMAKRISHNAN, C. R., AND SEKAR, R. Model-based analysis of configuration vulnerabilities. In *ACM CCS Workshop on Intrusion Detection Systems* (2000).

[7] RAMAKRISHNAN, C. R., AND SEKAR, R. Model-based analysis of configuration vulnerabilities. In *Journal of Computer Security* (2002).

[8] SEKAR, R., BENDRE, M., BOLLINENI, P., AND DHURJATI, D. A fast automaton based approach for learning program behaviors. In *IEEE Symposium on Security and Privacy* (2001).

[9] SEKAR, R., GUANG, Y., VERMA, S., AND SHANBHAG, T. A high-performance network intrusion detection system. In *ACM Symposium on Computer and Communication Security* (1999).

[10] SEKAR, R., RAMAKRISHNAN, C. R., RAMAKRISHNAN, I. V., AND SMOLKA, S. A. Model-carrying code (mcc): A new paradigm for mobile-code security. In *New Security Paradigms Workshop (NSPW'01)* (Cloudcroft, New Mexico, Sept 2001).

[11] SEKAR, R., AND UPPULURI, P. Synthesizing fast intrusion prevention/detection systems from high-level specifications. In *Proceedings of the USENIX Security Symposium* (1999).

[12] SEKAR, R., VENKATAKRISHNAN, V. N., BASU, S., BHATKAR, S., AND DUVARNEY, D. C. Model-carrying code: a practial approach for safe execution of untrusted applications. In *Procceedings of the 19th ACM Symposium on Operating Systems Principles* (October 2003).

[13] UPPULURI, P., AND SEKAR, R. Experiences with specification based intrusion detection system. In *Recent Advances in Intrusion Detection (RAID)* (October 2001).

[14] VENKATAKRISHNAN, V. N., RAM, P., AND SEKAR, R. Empowering mobile code using expressive security policies. In *Proceedings of the 10th New Security Paradigms Workshop* (Virginia Beach, VA, Sept 2002).

[15] VENKATAKRISHNAN, V. N., SEKAR, R., TSIPA, S., KAMAT, T., AND LIANG, Z. An approach for secure software installation. In *Proceedings of the 16th USENIX LISA conference, Philadelphia* (Nov 2002).